

ΣCLIPSE = Presentation Management + NASA Clips + SQL

Bernard P. Wess, Jr. *

May 1, 1990

Abstract

ΣCLIPSE provides a expert systems and "intelligent" data base development program for diverse systems integration environments that require support for automated reasoning and expert systems technology, presentation management, and access to "intelligent" SQL data bases. The ΣCLIPSE technology and its integrated ability to access 4th generation application development and decision support tools through a portable SQL interface, comprises a sophisticated software development environment for solving knowledge engineering and expert systems development problems in information intensive commercial environments—financial services, health care, and distributed process control—where the expert system must be extendable—a major architectural advantage of NASA Clips.

ΣCLIPSE is a research effort intended to test the viability of merging SQL data bases with expert systems technology.

1 Goals

ΣCLIPSE provides the Management Information Systems (MIS) expert systems developer a unique expert systems environment for:

Integration Much expert systems technology is too difficult for MIS developers to modify or does not adequately integrate with MIS data processing environments which demand support for corporate data bases and sophisticated visual presentation management facilities for professional and clerical users. [Schur88 Scown85] ΣCLIPSE supports commercial "mission critical" and "strategic" applications by extending NASA Clips to take advantage of Presentation Management (PM) functions and ANSI SQL data base access to enhance existing enterprise files and data bases. [Date88]

Standards ΣCLIPSE comprises functional extensions for portable text screens, windowing, fields, and menu development on a variety of operating systems and full graphics capabilities for the IBM PC under MSDOS and Borland Turbo C BGI graphics. ANSI SQL data base management is provided through an SQL C interface to a variety of file and data managers. [MIS89] ΣCLIPSE is an *extended*, not modified version of NASA Clips Version 4.3 [Giarr89] and includes objects and Clips facts and rules language source code for defining and manipulating—windows, forms, screens, reports, menus, fields, and icons.

*The author may be reached at Mentor Communications Ltd, 790 Highland Avenue, Needham Heights, MA 02194, (617)449-0086, Fax (617)449-0476.

2 PRESENTATION VERSUS DATA MANAGEMENT

Power Σ ECLIPSE provides high performance expert systems development capabilities to MIS professionals who need continuing compatibility with future NASA Clips upgrades, portable text windowing, and IBM PC graphics capabilities. All text-based presentation management is portable and IBM PC graphics-based presentation management is MSDOS "extended" to support 16 megabyte Clips applications for the Intel 386/486 processors.

Compatibility Σ ECLIPSE offers a complete implementation of NASA Clips and ANSI standard SQL including automatic ROLLBACK and COMMIT functions for commercial transaction processing. SQL is the only ANSI standard relational language for query, data manipulation, data definition and security. Applications developed IBM's SQL/DS and DB2 are very similar to Σ ECLIPSE SQL which is ANSI compatible.

Portability Text-based presentation management applications developed with Σ ECLIPSE can easily be ported to a wide-range of operating systems and computers, including: IBM PC/OS/2, Unix, DEC/VAX VMS. No Clips source code changes are required. SQL access to data bases and file managers is transparent within Σ ECLIPSE Clips rules and provides the application developer with the widest possible range of data retrieval and storage means, including: dBASE, Btrieve, C-tree, CB-tree, and in the future VAX/VMS Rdb, RMS, and Oracle.®

Systems Integration Future access to Oracle's distributed architecture (SQL*Connect), DEC DECNET, or TCP/IP Σ ECLIPSE applications will enable *distributed* Clips knowledge bases, when distributed processing is enabled within Clips, to reside on multiple computers and to access DBMS relations transparently through distributed SQL remote procedure calls. [Symb90 Adler90]

Architectural Freedom Σ ECLIPSE is divided into three layers—the User Front-End (Presentation Manager) in text or graphics modes, the Clips expert system compiler, Clips language, and the Back-End SQL data base engine implemented as both a Clips external function and internal Clips rule. All functional Σ ECLIPSE layers are independent architectural code layers which can be supplemented or replaced based on the changing requirements of the NASA Clips community of users and the demands of commercial MIS users.

2 Presentation versus Data Management

This section outlines the "front-end" of the Σ ECLIPSE product. Additionally, Σ ECLIPSE provides a dynamically re-configurable data base "back-end" which supports multiple data bases, platforms, applications, and communications environments based on SQL relational data bases. [MIS90]

The independence between presentation management and back-end data base management is provided by a data object object, the actions applied to an object, its relationship(s) to other object(s), and the screen representation of the object from access, through SQL to underlying components or sub-objects of the parent object. [Shu89] Thus an unlimited range of presentation metaphors can be used to represent user interactions. The Σ ECLIPSE front-end enables a vastly expanded level of functionality to be incorporated in the presentation, display, manipulation and interaction between application and screen processes and the user. The level of complexity of screen presentation and interaction is greatly enhanced over existing front-ends which either:

- rely on a single metaphor for interacting with the user or display of visual objects or

3 ARCHITECTURAL OVERVIEW

- require that the front-end be used to build complex displays through the use of icons.

These methods do not enable the use of more complex forms of interaction to be integrated or enhanced in the graphic front-end.

3 Architectural Overview

ECLIPSE has the following characteristics:

Object-oriented “Intelligent” displays are comprised of visual objects that have meaning and actions associated with them. Screens are built from complex objects and icons and their associated actions, predetermined by the user/develop or the application logic, lead the user through an application. Icons and/or complex objects (for example, data base tables or spreadsheets) can be moved, manipulated, or act as triggers when activated or changed.

Active Screen Metaphors Any graphical metaphor can be set up which makes sense to the user and aids in representing the underlying application logic or data base. For example, a manager may interact with the program through a spreadsheet where each cell is an active object that itself may be another spreadsheet or piece of a data base. Or a screen may represent a chemical processing plant from which the user can control the operations by manipulating dials, meters, switches, etc.

Virtual Objects The physical screen is not a limitation on the size of a screen object and the data or image it represents. For example, a spreadsheet could handle large data base tables of virtually unlimited size with numerous graphs located in cells as associated screen “child” objects. Or a physical window may represent only a portion of a larger graphical image and the image may be zoomed, panned, expanded, etc.

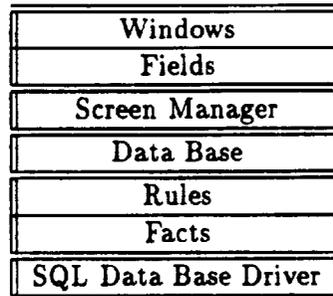
Ease of Use The user interface is highly intelligent, intended for use by professional managers and office workers as well as by MIS professionals.

Integrated DBMS Display tools, such as forms builders and spreadsheets may be integrated in a “seamless” manner through the definition of more complex objects.

Flexibility Because of the modular nature of the product, design flexibility and independence of architecture, interfaces among ECLIPSE modules, underlying application programs, communications technology, and the data base manager are easily modified or replaced. Any component of the program or any associated application can be replaced by another product, such as Excel in the spreadsheet arena or Sybase in the DBMS area.

The simple architecture is outlined in Figure 1.

Figure 1: The simple architecture of the ECLIPSE .



4 ECLIPSE Front-End Features

4.1 General Features

The *general features* of the the program include the following and are available regardless of the Front-end (FE) mode of operation. Both text and graphics modes of operation can be executed simultaneously.

FE Drivers The FE driver can be replaced or enhanced to extend the functionality of the FE.

C Interface The user's application and the FE itself may be extended or merged by *registering* external C functions to be recognized by ECLIPSE and/or the user's application.

Compiled Screens All screen objects—windows and fields, for example, are compiled once at application run-time if they are not loaded in binary format. Therefore, screen objects are manipulated on the display with maximum speed so the user sees fast screen updates.

4.2 The Text-based Front-End Features

The text-based FE provides the following functionality:

4.2.1 Window & Display Control

Windows A window is an area of the *logical* screen that is treated as a separate display entity. Windows may have borders, overlap, or cover one another and have a priority that is user or application assigned in real-time. They may also be larger then the physical display.

Forms or Pages A "page" of "forms" is a virtual screen which may be smaller, larger, or the same size as a physical display. Pages or windows may be named.

Scrolling Automatic scrolling is accomplished to orient the proper current window. Scrolling may be horizontal or vertical, as required. The application may write to hidden window areas without causing scrolling.

Logical Write An application may write to a window without causing a window to update the physical screen until all application output has ended.

4 ECLIPSE FRONT-END FEATURES

Logical Attributes A display "attribute" table is maintained which is logical in nature. The logical display attributes, for example, "red" are converted to display driver output in real-time by the FE.

Character Writing Characters may be written with or without attributes and for one or more characters.

String Writing Strings may be written with or without attributes and for one or more characters.

Cursor Control Full "virtual" cursor control is available including the ability to write to a window's "current position," with a virtual or physical row and column attached. The cursor may "drag" the virtual windows, leave the display unchanged, or make the cursor position the current position.

4.3 Field Level Functions

The following functions are available for fields:

Repeating Lines A formatted "block" may be entered once and repeated for scrolling formatted windows of identical lines of input. The lines in a block can exceed the actual physical window size and automatic scrolling will occur.

Validation Format strings control the character field-level input and output. ECLIPSE rules can intercede to more fully control field and character I/O based on external application function calls or rule execution.

Field Editing A full-featured text editor is automatically invoked for each field. Intra- and extra-field movement can be controlled within a window or field.

Field Functions Window and field level functions include exit to next field or previous field, field above or field below, beginning or end of window, previous or next window, line up or down, send data, delete data, or abort.

Field Data Types String; 8-, 16-, 32-bit signed and unsigned binary, 32-bit monetary, date, time, 32- and 64-bit floating point.

Edit Patterns Character strings format a field on input or output so that formatted fields are properly presented, for example, account, SSNs and telephone numbers.

4.4 Graphical User Interface Functions

The following functions are available within the GUI and are unique to the bit-mapped GUI FE:

Graphics System Automatic detection of hardware and resolution and driver loading for more than 30 modes of operation. Movement from character-based I/O to bit-mapped graphics is supported. Multiple pages of graphics are supported for the appropriate hardware drivers.

Graphics An unlimited variety of objects can be drawn directly on the physical screen (not in window buffers). Arcs, circles, polygons, ellipses, lines, points, 3-D and 2-D bars and bar

5 FRONT-END COMMUNICATIONS INTERFACE

charts, line and point charts, pie slices, rectangles, as well as icons of any size are available. Functions to flood and pattern fill objects, rotate, zoom, move and manipulate lines and polygons are available.

Fonts & Icons Multiple fonts are available, including *sans-serif*, gothic, triplex, and roman. Fonts may be oriented, sized, colored, and transposed. Icons may be loaded and displayed from external font and icon tables. Pixels, characters, strings, and images may be interrogated or manipulated. Both bit-mapped and "stroked" fonts are available.

Graphic Function Library Many internal functions, accessible to user processes, programs, and functions are provided for business and engineering graphics.

Text Output Full-text control is available including style, centering, color, orientation, size, and magnification.

Color Control Color can be applied through a "color palette" to objects, characters, windows, and pixels. A "color table" is defined to control colors.

State Control The "state" of an object, character, window, display or pixel can be interrogated and the results sent as a message as a fact into the data base or to a function. Full application control is enabled through the message system to maintain flexibility.

Icon/Object Library Graphical objects may be created with the graphics editor and dynamically called from memory or disk. A library of icons and presentation management metaphors is available for customization and use in new user defined applications.

5 Front-End Communications Interface

The FE utilizes the Σ CLIPSE development language and support utilities to "define" or to "create" display objects such as windows and fields. Moreover, Σ CLIPSE can manipulate any defined object. The FE or the application can *directly* execute SQL commands.

Σ CLIPSE allows any data base or graphical object to be modified by sending command *messages* to Σ CLIPSE from user-developed applications, external events, and changes in the state of objects or data. Also, objects may send messages directly to other objects for processing, without the need for application program or user intervention.

6 REFERENCES

6 References

[Schur88] Stephen Schur, "The Intelligent Data Base", *AI Expert*, pp. 26-34, Jan. 1988

[Scown85] Susan B. Scown, *The Artificial Intelligence Experience: An Introduction*, Digital Equipment Corporation, 1985.

[Date88] Date & White, *A Guide to DB2*, Second Edition, Addison Wesley, New York, 1988.

[MIS89] *CQL Data Base/Program Development System*, Version 6.0, Machine Independent Software, Fourth Edition, 1989.

[Giarr89] Giarratano & Riley, *Expert Systems: Principles and Programming*, PWS-Kent, Boston, MA, 1989.

[Symb90] Symbiotics, *MetaCourier User's Guide*, Symbiotics, Inc., Jan. 1990, Pre-Release Version 1.2.4, Cambridge MA 1990.

[Adler90] Adler & Cottman, *A Development Framework for AI Based Distributed Operations Support Systems*, Fifth Conference on AI for Space Applications, Huntsville, Alabama, May 22-23, 1990

[Shu89] Shu, N., "Visual programming: Perspectives and approaches", *IBM Systems Journal*, Vol. 28, No. 4, 1989, pp. 525-547.

CLIPS1.TEX